

## ДОРАБОТКА ИМИТАЦИОННОЙ МОДЕЛИ АЛГОРИТМОВ ПРИОРИТИЗАЦИИ В СЕТЯХ ТСП/П

Р.Х. Ниязов, Ю.М. Монахов, И.С. Бедняцкий, В.И. Балашов, А.П. Кузнецова (Владимир)

В настоящее время основой работы глобальной сети Интернет являются протоколы ТСП/П — это стек протоколов, по которому происходит обмен информации в приложениях и сервисах. Их можно «грубо» разделить по чувствительности к задержкам на два класса: асинхронные и синхронные. К асинхронным относятся те приложения, которые нечувствительны к задержкам передачи данных в диапазоне до нескольких секунд, а все остальные приложения относят к синхронным, на функциональность которых сильно влияют задержки.[3, 4]

Использование алгоритмов приоритизации трафика позволяет отделить синхронный трафик и выделить ему гарантированную полосу пропускания. Таким образом, сейчас актуальна задача разработки алгоритмов приоритизации трафика, позволяющих обеспечить синхронному трафику допустимые задержки.[3, 4]

Алгоритм НТВ. Данный алгоритм соответствует требованиям для обработки синхронного трафика. Классы могут разделяться на дочерние классы, каждый из которых будет делить между собой полосу родительского класса. Каждый класс соответствует определенному типу трафика и имеет свой приоритет.

Обозначим основные параметры необходимые для работы алгоритма:  $\max\text{-limit}$  (максимальная скорость, которую можно предоставить классу);  $\text{limit-at}$  (минимальная гарантированная скорость, которой необходимо обеспечить классу);  $\text{priority}$  (приоритет класса). Когда несколько классов претендуют на канал родительского, они получают части канала пропорционально их квантам.[5, 6]

Фундаментальной частью НТВ  $\text{qdisc}$  является механизм заимствования. Дочерние классы берут токены у своих родителей, как только они превысят  $\text{limit-at}$ . Дочерние классы будут продолжать заимствование, пока не достигнут  $\text{max-limit}$ , после чего класс начнет помещать в очередь пакеты для передачи, пока не будет доступно больше токенов.[5,6]

Чтобы модель заимствования работала, каждый класс должен иметь точный подсчет количества используемых токенов самим собой и всеми своими детьми. По этой причине любой токен, используемый в дочернем или листовом класс, взимается с каждого родительского класса, пока не будет достигнут корневой класс.[5, 6]

Заимствование токенов направленно на листовые классы и взимание платы за использование токенов ведет к корневому классу. Шейпинг происходит только в листовых классах.

В данном докладе будет модифицирован алгоритм НТВ: при заимствовании токенов алгоритм будет опираться на соблюдение вхождения задержки класса директивному интервалу времени.

Формализация цели. Трафик необходимо группировать по приоритетам в соответствии с соглашением о качестве обслуживания. Деление полосы пропускания будет происходить в соответствии с соблюдением заявленного допустимого времени ожидания высокоприоритетного трафика. [5]

Определим множество  $O$  – как множество сервисов, где каждая  $o_i$  – является  $i$ -сервисом. [1]

$$O = \{o_1, o_2, \dots, o_n\} (1)$$

Определим множество  $T$  – как множество откликов сервисов, где каждая  $t_i$  – является временем отклика  $i$ -того сервиса. [1]

$$T = \{t_1, t_2, \dots, t_n\} (2)$$

Определим множество  $S$  – как множество соглашения об обслуживании (SLA), где каждая  $s_i$  – максимальное время отклика для каждого  $i$ -того сервиса: [1]

$$S = \{s_1, s_2, \dots, s_n\} (3)$$

### Секция 3. Практическое применение моделирования и инструментальных средств автоматизации моделирования, принятие решений по результатам моделирования

---

Определим множество  $Q$  – множество классов, где  $q_j$  состоит из нескольких сервисов: [1]

$$Q = \{q_1, q_2, \dots, q_n\}, \text{ где } q_j \subset 0(4)$$

Определим множество  $C$  – как множество приоритетов каждого класса, где каждая  $c_j$  – является приоритетом  $j$ -того класса, где класс с наименьшим значением  $c$  имеет наивысший приоритет: [1]

$$C = \{c_1, c_2, \dots, c_m\}, \text{ где } c_j \in [0, j](5)$$

Сами классы группируются таким образом, что сервисы, оказавшиеся в одном классе, имеют схожее значение  $s$ . Преимущество в обработке будут иметь высокоприоритетные классы. Тем самым, полоса пропускания будет делиться на классы в соотношении, в котором, чем приоритетней класс, тем больше полос пропускания он получит.

Описание работы модели. В рамках алгоритма производится классификация трафика по определенным признакам, таким как: IP-адрес назначения или источника, порт назначения или источника, протокол передачи данных и т.д.

Алгоритм имеет одну очередь передачи в среду для пакетов различных классов. Из очередей классов пакеты выходят с интенсивностью меньше или равной пороговой интенсивности для этого класса. Пакеты попадают в очередь последовательно в соответствии с их приоритетами.

Изначально большая часть полосы пропускания будет разделена на гарантированные участки для каждого класса. Деления полосы пропускания будет происходить в соответствии с соблюдением заявленного допустимого времени ожидания для трафика и приоритетов. Это означает, что алгоритм будет пытаться соблюдать заявленную допустимую задержку для высокоприоритетного класса. В случае, если высокоприоритетный класс укладывается в установленное директивное время, то появляется возможность отдать оставшуюся полосу пропускания менее приоритетному классу.

Если задержка класса больше установленного директивного времени, то он будет пытаться подниматься вверх по дереву алгоритма, заимствовав больше полосы пропускания до тех пор, пока задержка не станет удовлетворительной. В случае, когда текущая задержка класса гораздо меньше заявленного директивного времени, класс будет отдавать заимствованную им полосу пропускания, если такая имеется.

Моделирование алгоритма. Имитация описанной модели производилась в среде моделирования AnyLogic.

В алгоритме НТВ листьями являются очереди. А выше стоит корневой класс. Посередине между ними могут располагаться промежуточный класс. Сами пакеты приходят на обработку только в листья. [1]

Состав моделируемой системы

Моделируемая среда состояла из:

- Очередь передачи пакетов с приоритетом 0 и максимальным директивным временем 2 мс;
- Очередь передачи пакетов с приоритетом 1 и максимальным директивным временем 5 мс;
- Очередь передачи пакетов с приоритетом 2 и максимальным директивным временем 10 мс;
- Планировщик распределения использования очереди передачи в среду для очередей с приоритетами.[1]

Каждая моделируемая очередь обладала следующими параметрами:

- Интенсивность потока пакетов, *flow<sub>i</sub>*;
- Гарантированная полоса пропускания листа, *limit\_at\_leaf<sub>i</sub>*;
- Время обработки пакетов в очереди, *delay<sub>i</sub>*;
- Максимальная задержка пакетов из очереди за последнее время, *last\_max\_delay<sub>i</sub>*. [1]

Для очередей задавались следующие ограничения:

- Максимальная полоса пропускания, которую можно заимствовать листу, *max\_limit\_leaf<sub>i</sub>*;
- Буфер очередей с приоритетами был ограничен в значение равное 150 пакетов;

### Секция 3. Практическое применение моделирования и инструментальных средств автоматизации моделирования, принятие решений по результатам моделирования

- Время обработки пакетов в очереди передачи пакетов в среду передачи, всегда составляет  $1/\max\_flow$ ;[1]

Так же в модели реализован класс В, который является потомком первой и третьей очереди. Для него были так же установлены значения  $limit\_at\_classB$  и  $max\_limit\_classB$ . Корневой класс А реализован в виде максимальной ширины полосы пропускания,  $max\_flow$ . [1]

Модель постоянно подсчитывает максимальную задержку вышедших пакетов для каждого приоритета за последнее время и на основе данной информации производит манипулирование полосой пропускания. Заимствования полосы пропускания происходят на основе иерархического дерева (Рисунок 2).

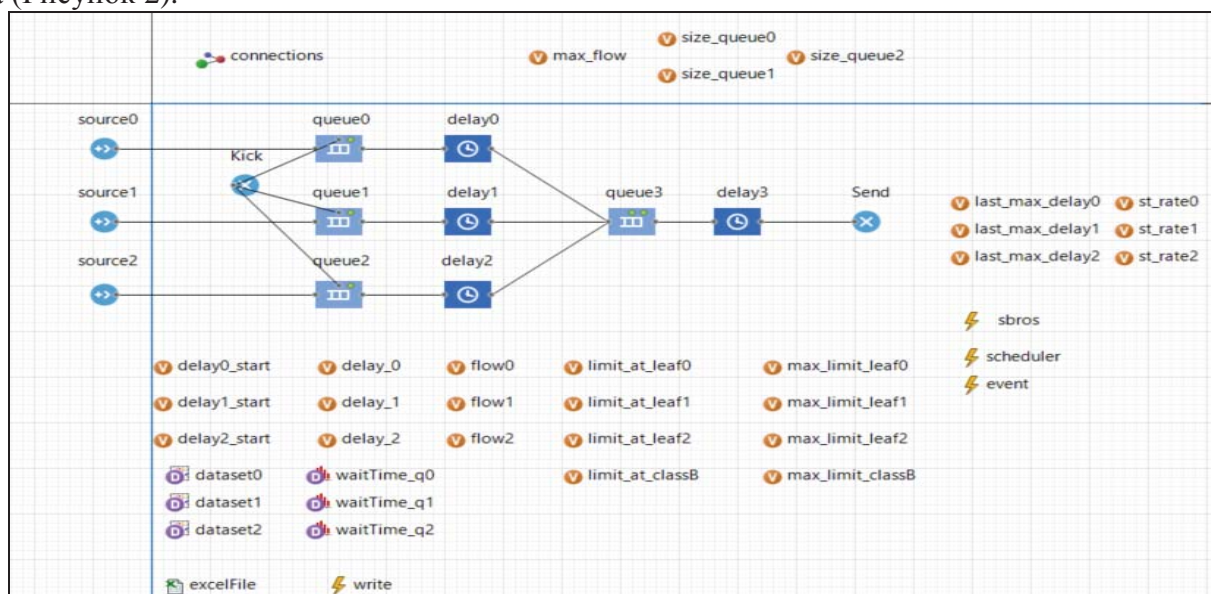


Рис.1 Имитационная модель

#### Тестирование модели

Для определения эффективности построенной модели планирования очередей необходимо провести исследование. Исследование предполагает 6 тестовых сьемов задержки передачи пакетов в различных условиях. Режимы тестирования представлены в таблице 1. Каждый тест проводился 2 минуты, и отмечалась задержка для каждого пакета.[2]

Таблица 1 – Режимы тестирования модели

Режим тестирования	Интенсивность очереди 1 (пакет/мс)	Интенсивность очереди 2 (пакет/мс)	Интенсивность очереди 3 (пакет/мс)
Режим 1	5	5	5
Режим 2	5	5	50
Режим 3	5	50	5
Режим 4	50	5	50
Режим 5	5	50	50
Режим 6	50	50	50

#### Результаты тестирования

Результаты проведенного эксперимента представлены в виде графиков. На рисунке 2 представлен график работы алгоритма в режиме 1. Из графика видно, что при низкой интенсивности входящего потока пакетов обеспечивается низкая задержка обработки пакетов. Однако для низко приоритетной очереди, задержка выше приоритетного класса, это обусловлено последовательностью права передачи пакетов на основании приоритета.

### Секция 3. Практическое применение моделирования и инструментальных средств автоматизации моделирования, принятие решений по результатам моделирования

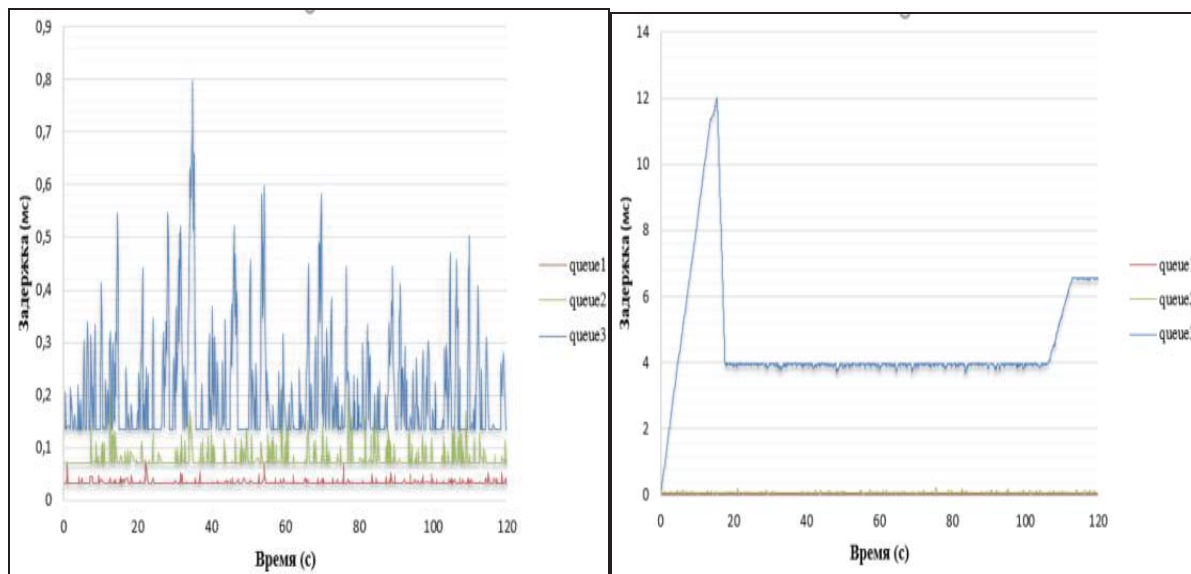


Рис.2, 3 - Результат тестирования модели в режиме 1, 2

На рисунке 3 представлен график работы алгоритма в режиме 2. На графике видно, что алгоритм выделил оставшуюся полосу пропускания для третьей очереди, так как задержка её пакета преодолела допустимую. Третья очередь сначала заняла класс В, а потом и класс А. Алгоритм позволил это сделать, так как, более приоритетный трафик не нуждался в дополнительной полосе.

На рисунке 4 представлен график работы алгоритма в режиме 3. На графике видно, что алгоритм выделил часть свободной полосы пропускания для второй очереди, так как задержка её пакета преодолела допустимую. Вторая очередь сразу заняла класс А, так как она не связана с классом В. Свободная часть полосы пропускания В не доступна для второй очереди. Алгоритм позволил занять второй очереди класс А, так первая очередь была разгружена.

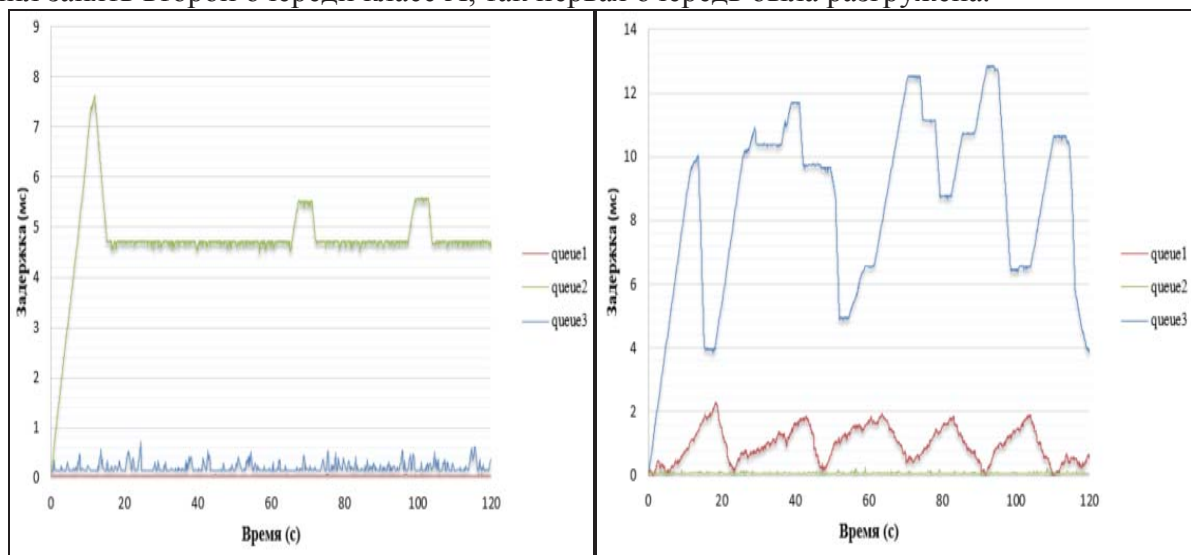


Рисунок 4, 5 - Результат тестирования модели в режиме 3, 4

На рисунке 5 представлен график работы алгоритма в режиме 4. Так как обе очереди подключены к одному классу В, то они являются прямыми конкурентами. На графике видно, что алгоритм отдает полосу пропускания первой очереди, когда директивное время приближается к максимально допустимому, и забирает полосу, когда очередь разгружается. Очередь 1 поднимается в класс А через класс В, а потом опускается обратно. В то время, когда первая



### Секция 3. Практическое применение моделирования и инструментальных средств автоматизации моделирования, принятие решений по результатам моделирования

очередь разгружена, алгоритм предоставляет свободную полосу третьей очереди. Она также поднимается в класс А через класс В, и опускается в локальную очередь, как только первой очереди снова потребуется полоса.

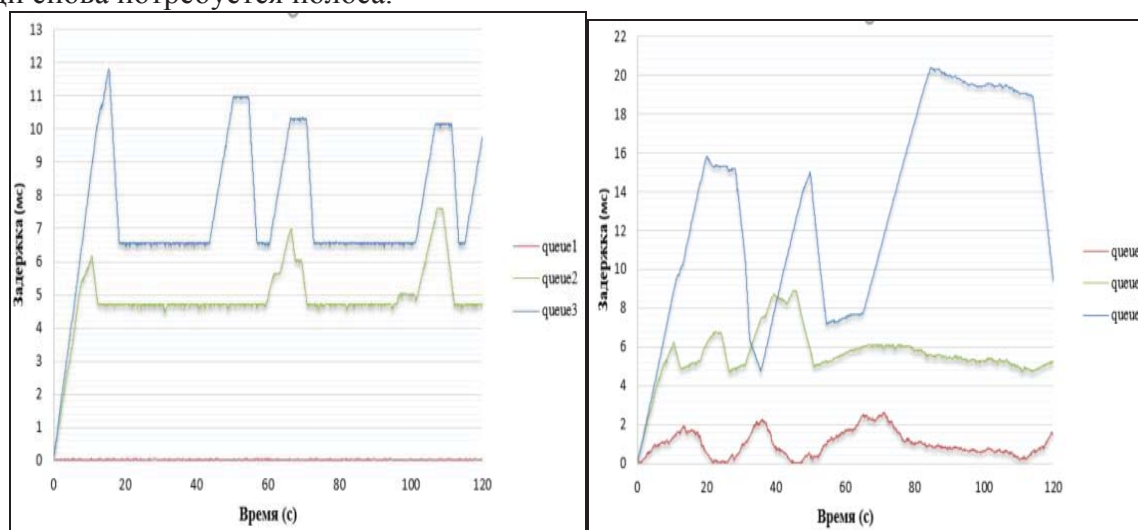


Рисунок 6, 7 - Результат тестирования модели в режиме 5, 6

На рисунке 6 представлен график работы алгоритма в режиме 5. На графике видно, что алгоритм отдает часть полосы пропускания второй очереди и часть – третьей очереди, когда их задержки превысили максимальное директивное время. Это получается потому, что вторая очередь не связана с классом В. Данный факт позволяет занять класс В третьей очереди. Сама очередь 2 напрямую подключается к классу А. Таким образом, обеим очередям предоставляется часть свободной полосы пропускания, за счёт построения модели.

На рисунке 7 представлен график работы алгоритма в режиме 6. Изначально, когда задержки во всех очередях максимальны к допустимым, алгоритм отдает полосу пропускания первой очереди, так как она является высокоприоритетной, и затем отбирает у неё полосу как в режиме 4. Когда первая очередь является разгруженной, свободная полоса пропускания делится между второй и третьей очередью как в режиме 5. В том случае, если задержка первой очереди приближается к максимальному директивному времени, то заимствованная полоса отбирается у второй и третьей очереди и полностью отдается очереди один.

Результаты проведения тестов генерации трафика

Первый способ генерации трафика. В данном случае класс трафика с большим приоритетом генерируется в соответствии с нормальным распределением, а два класса меньшего приоритета распределением Парето.

Таблица 2 - Первый способ генерации трафика

Тест		№1	№2	№3	№4	№5
Класс С	Количество пакетов на входе	9965	9807	9 835	9947	9986
	Пакеты не прошедшие за мдв (%)	<b>1.18</b>	<b>2.32</b>	<b>23,57</b>	<b>4,60</b>	<b>61,64</b>
	Средняя задержка	0.5	0.82	1,37	0,56	2,30
Класс 1	Количество пакетов на входе	4623	4850	4 570	4241	4568
	Пакеты не прошедшие за мдв (%)	<b>1.576</b>	<b>3.91</b>	<b>0,57</b>	<b>5,28</b>	<b>13,38</b>
	Средняя задержка	0.86	1.57	1,54	1,07	2,57
Класс 2	Количество пакетов на входе	2912	3.164	3 617	3531	3804
	Пакеты не прошедшие за мдв (%)	<b>5.66</b>	<b>5.05</b>	<b>3,23</b>	<b>6,65</b>	<b>5,28</b>
	Средняя задержка	1.9	2.57	2,09	2,41	2,86

### Секция 3. Практическое применение моделирования и инструментальных средств автоматизации моделирования, принятие решений по результатам моделирования

При первом способе генерации трафика можно увидеть, что алгоритм не всегда выполняет свои задачи при перегрузках близких к предельным.

Второй способ генерации трафика. В этом способе трафик генерируется также, как и в первом, но к каждому прибавлена переменная, изменяющаяся во времени.

Значение переменной максимально в начале и конце эксперимента и минимально в середине.

Таблица 3 - Второй способ генерации трафика

		Тест	№1	№2	№3	№4	№5
Класс с	Количество пакетов на входе		9053	9637	9193	9026	9086
	Пакеты не прошедшие за мдв (%)		<b>7,94</b>	<b>10,00</b>	<b>11,61</b>	<b>26,81</b>	<b>35,04</b>
	Средняя задержка		0,71	0,83	1,19	1,41	1,69
Класс 1	Количество пакетов на входе		4227	4246	4643	5361	4815
	Пакеты не прошедшие за мдв (%)		<b>4,19</b>	<b>1,74</b>	<b>11,65</b>	<b>38,65</b>	<b>5,59</b>
	Средняя задержка		1,31	1,26	2,49	4,03	2,30
Класс 2	Количество пакетов на входе		3792	3572	4063	3939	4603
	Пакеты не прошедшие за мдв (%)		<b>1,34</b>	<b>4,59</b>	<b>4,18</b>	<b>3,33</b>	<b>3,08</b>
	Средняя задержка		1,72	2,50	3,39	2,82	4,63

При одновременном одинаковом увеличении интенсивности поступления пакетов во всех классах алгоритм проявляет себя хуже, но ведет себя схоже.

Третий способ генерации трафика. В данном случае класс трафика с большим приоритетом генерируется в соответствии с нормальным распределением, класс меньшего приоритета распределением Парето, а оставшийся генерируется в соответствии с определенным периодом.

Таблица 4 – Третий способ генерации трафика №1

		Тест	№1	№2	№3	№4	№5
Класс с	Количество пакетов на входе		9002	9793	9766	9628	9742
	Пакеты не прошедшие за мдв (%)		<b>23,54</b>	<b>10,30</b>	<b>2,08</b>	<b>66,77</b>	<b>57,94</b>
	Средняя задержка		1,40	1,03	0,78	2,71	2,38
Класс 1	Количество пакетов на входе		5009	4725	4725	4725	4725
	Пакеты не прошедшие за мдв (%)		<b>35,24</b>	<b>11,17</b>	<b>7,75</b>	<b>29,95</b>	<b>27,83</b>
	Средняя задержка		4,09	3,08	2,91	3,98	3,90
Класс 2	Количество пакетов на входе		5241	3341	3405	3817	3766
	Пакеты не прошедшие за мдв (%)		<b>2,46</b>	<b>3,62</b>	<b>3,55</b>	<b>4,22</b>	<b>4,12</b>
	Средняя задержка		6,91	1,98	2,05	2,76	2,80

В нулевом и первом классах стало появляться ещё больше пакетов, не прошедших за мдв. В классе с периодическим трафиком процент находится в окрестностях 30.

В следующих тестах класс высшего приоритета - периодический, а среднего с нормальным распределением.

### Секция 3. Практическое применение моделирования и инструментальных средств автоматизации моделирования, принятие решений по результатам моделирования

Таблица 5 - Третий способ генерации трафика №2

		Тест	№1	№2	№3	№4	№5
Класс	Количество пакетов на входе		9230	9620	9620	9230	9230
	Пакеты не прошедшие за мдв (%)		<b>33,55</b>	<b>69,69</b>	<b>42,71</b>	<b>37,58</b>	<b>63,28</b>
	Средняя задержка		1,61	2,61	1,86	1,73	2,41
Класс 1	Количество пакетов на входе		4828	4957	4958	4703	4872
	Пакеты не прошедшие за мдв (%)		<b>6,15</b>	<b>63,08</b>	<b>2,32</b>	<b>0,81</b>	<b>15,89</b>
	Средняя задержка		1,37	5,43	1,38	1,35	3,65
Класс 2	Количество пакетов на входе		3771	3756	3496	3839	3778
	Пакеты не прошедшие за мдв (%)		<b>6,52</b>	<b>4,02</b>	<b>7,41</b>	<b>2,71</b>	<b>0,13</b>
	Средняя задержка		2,72	2,56	2,74	2,91	2,54

Задержка пакетов в нулевом классе возросла. Количество не прошедших пакетов в первом классе в тестах 1,3,4 в пределах нормы, в тестах 2 и 5 63% и 16% соответственно. Во втором классе все значения в пределах нормы.

Вывод: Добавление периодического трафика большого объема нарушает работу алгоритма, особенно сильно влияет на нулевой класс.

#### Заключение

Результаты тестирования генерации трафика показали, что модификация алгоритма НТВ справляется с задачей распределения количества обрабатываемых пакетов по приоритету, но во многих случаях количество пакетов, не прошедших за максимальное директивное время, является слишком большим в нулевом классе. По результатам имитационного моделирования работы алгоритма видна зависимость работы алгоритма планирования очередей передачи данных от задержек пакетов в очередях.

#### Литература

1. Traffic Prioritization To Ensure Availability in Telecommunication Networks of Textile Enterprises/ Monakhov Yu.M., Kuznetsova A.P., Pestov A.V.// 06.06.2018
2. Дисциплина обработки очереди НТВ. Руководство по использованию [Электронный ресурс] 2006 г.// Алексей Ремизов // [https://www.opennet.ru/URL:https://www.opennet.ru/base/net/htb\\_manual.txt.html](https://www.opennet.ru/URL:https://www.opennet.ru/base/net/htb_manual.txt.html) (Дата обращения 15.08.2019)
3. Компьютерные сети [Текст] / В. Олифер, Н. Олифер // Питер 5-е издание 2016 г.
4. Основы информационных технологий IP-ТЕЛЕФОНИЯ в КОМПЬЮТЕРНЫХ СЕТЯХ/ И.В. Баскаков, А.В. Пролетарский, С.А. Мельников, Р.А. Федотов // Интернет-Университет Информационных Технологий, БИНОМ. Лаборатория знаний 2008 г.
5. Управление трафиком в Linux 2009г./infocity.kiev.uaURL: <http://www.infocity.kiev.ua/os/content/os395.phtml> (Дата обращения 16.07.2019)
6. Implementing Real Time Packet Forwarding Policies using HTB / Octavian V. RUSU, Manuel SUBREDU, Ionut SPARLEA, Valeriu VRACIU